Why BSD is better than Linux

Greg ‘‘groggy’’ Lehey
IBM Linux Technology Center, Ozlabs
grog@FreeBSD.org
grog@au1.ibm.com
Brisbane, 7 February 2002

BSD? What's that?

BSD? What's that?

Remember these?

vi

BSD? What's that?

Remember these?

vi

csh

BSD? What's that?

Remember these?

vi

csh

sendmail

BSD? What's that?

Remember these?

vi

csh

sendmail

named (BIND)

BSD? What's that?

Remember these?

vi

csh

sendmail

named (BIND)

The Internet

BSD? What's that?

Remember these?

vi

csh

sendmail

named (BIND)

The Internet

OpenSSH

A little history

A little history

- UNIX was a research project until 1982.

A little history

- UNIX was a research project until 1982.
- Universities had access to the UNIX source code.

A little history

- UNIX was a research project until 1982.
- Universities had access to the UNIX source code.
- The Computer Sciences Research Group (CSRG) at the University of California, Berkeley (UCB) wrote much new code.

A little history

- UNIX was a research project until 1982.
- Universities had access to the UNIX source code.
- The Computer Sciences Research Group (CSRG) at the University of California, Berkeley (UCB) wrote much new code.
- The code was released on tapes called ‘‘Berkeley Software Distribution’’, or BSD.

BSD in the 80s

BSD in the 80s

- UNIX was ported to the VAX, which required virtual memory support. The first widely distributed UNIX for the VAX was 3BSD.

BSD in the 80s

- UNIX was ported to the VAX, which required virtual memory support. The first widely distributed UNIX for the VAX was 3BSD.
- The American Defense Advanced Research Projects Administration (DARPA) wanted to update their network, ARPANET.

BSD in the 80s

- UNIX was ported to the VAX, which required virtual memory support. The first widely distributed UNIX for the VAX was 3BSD.
- The American Defense Advanced Research Projects Administration (DARPA) wanted to update their network, ARPANET.
- The new protocols were called ‘‘Internet Protocols’’.

BSD in the 80s

- UNIX was ported to the VAX, which required virtual memory support. The first widely distributed UNIX for the VAX was 3BSD.
- The American Defense Advanced Research Projects Administration (DARPA) wanted to update their network, ARPANET.
- The new protocols were called ‘‘Internet Protocols’’.
- DARPA gave the contract to BBN and UCB.

BSD in the 80s

- UNIX was ported to the VAX, which required virtual memory support. The first widely distributed UNIX for the VAX was 3BSD.
- The American Defense Advanced Research Projects Administration (DARPA) wanted to update their network, ARPANET.
- The new protocols were called ‘‘Internet Protocols’’.
- DARPA gave the contract to BBN and UCB.
- The first operating system to support the Internet Protocols was 4.1c BSD.

But is it free software?

But is it free software?

- UNIX was not free software.

But is it free software?

- UNIX was not free software.
- All work done at UCB was available without licensing conditions.

But is it free software?

- UNIX was not free software.
- All work done at UCB was available without licensing conditions.
- The tapes contained both UNIX and UCB code, so they could only go to UNIX license holders.

But is it free software?

- UNIX was not free software.
- All work done at UCB was available without licensing conditions.
- The tapes contained both UNIX and UCB code, so they could only go to UNIX license holders.
- In the late 80's, the CSRG worked to extricate the Berkeley code from the UNIX code.

But is it free software?

- UNIX was not free software.
- All work done at UCB was available without licensing conditions.
- The tapes contained both UNIX and UCB code, so they could only go to UNIX license holders.
- In the late 80's, the CSRG worked to extricate the Berkeley code from the UNIX code.
- The results were released as the ''Berkeley Networking Tapes'', Net/1 and Net/2.

But is it free software?

- UNIX was not free software.
- All work done at UCB was available without licensing conditions.
- The tapes contained both UNIX and UCB code, so they could only go to UNIX license holders.
- In the late 80's, the CSRG worked to extricate the Berkeley code from the UNIX code.
- The results were released as the ''Berkeley Networking Tapes'', Net/1 and Net/2.
- They were not complete operating systems.

A free UNIX operating system

- In the early 90s, Bill Jolitz ported 4.3BSD to the Intel 386 architecture.

A free UNIX operating system

- In the early 90s, Bill Jolitz ported 4.3BSD to the Intel 386 architecture.
- Some old CSRG members formed a company called Berkeley Software Design, Inc. (BSDI).

A free UNIX operating system

- In the early 90s, Bill Jolitz ported 4.3BSD to the Intel 386 architecture.
- Some old CSRG members formed a company called Berkeley Software Design, Inc. (BSDI).
- BSDI marketed a commercial operating system called BSD/386, later BSD/OS.

A free UNIX operating system

- In the early 90s, Bill Jolitz ported 4.3BSD to the Intel 386 architecture.
- Some old CSRG members formed a company called Berkeley Software Design, Inc. (BSDI).
- BSDI marketed a commercial operating system called BSD/386, later BSD/OS.
- Bill Jolitz wanted a free version, and created 386BSD.

A free UNIX operating system

- In the early 90s, Bill Jolitz ported 4.3BSD to the Intel 386 architecture.
- Some old CSRG members formed a company called Berkeley Software Design, Inc. (BSDI).
- BSDI marketed a commercial operating system called BSD/386, later BSD/OS.
- Bill Jolitz wanted a free version, and created 386BSD.
- Later, other people used this basis to create NetBSD (April 1993), FreeBSD (December 1993) and OpenBSD (October 1995).

But is it UNIX?

But is it UNIX?

- UNIX is a trade mark of AT&T.

But is it UNIX?

- UNIX is a trade mark of AT&T.
- AT&T is a modem test command.

But is it UNIX?

- UNIX is a trade mark of The Open Group.

But is it UNIX?

- UNIX is a trade mark of The Open Group.
- The Open Group awards licenses after proving specific functionality and purchasing power.

But is it UNIX?

- UNIX is a trade mark of The Open Group.
- The Open Group awards licenses after proving specific functionality and purchasing power.
- The BSDs have not purchased a UNIX name.

But is it UNIX?

- UNIX is a trade mark of The Open Group.
- The Open Group awards licenses after proving specific functionality and purchasing power.
- The BSDs have not purchased a UNIX name.
- Microsoft has purchased a UNIX name for ‘‘Windows NT’’.

But is it UNIX?

- UNIX is a trade mark of The Open Group.
- The Open Group awards licenses after proving specific functionality and purchasing power.
- The BSDs have not purchased a UNIX name.
- Microsoft has purchased a UNIX name for ‘‘Windows NT’’.
- The Eighth edition of AT&T Research UNIX was based on 4.1cBSD.

But is it UNIX?

- UNIX is a trade mark of The Open Group.
- The Open Group awards licenses after proving specific functionality and purchasing power.
- The BSDs have not purchased a UNIX name.
- Microsoft has purchased a UNIX name for ‘‘Windows NT’’.
- The Eighth edition of AT&T Research UNIX was based on 4.1cBSD.
- To be sure, look at the code.

But is it UNIX?

Seventh Edition Research UNIX *sys/dev/rl.c* :

```
rlstrategy(bp)
register struct buf *bp;
{
        register struct rl *rlp;
        int drive,dsize;

        drive = minor(bp->b_dev);
        rlp = &rl[drive];
        dsize = 0;
...
        spl5();
        if(rltab.b_actf == NULL)
                rltab.b_actf = bp;
        else
                rltab.b_actl->av_forw = bp;
        rltab.b_actl = bp;
        if(rltab.b_active == NULL)
                rlstart();
        spl0();
}
```

But is it UNIX?

FreeBSD *sys/i386/isa/wd.c*:

```
void
wdstrategy(register struct buf *bp)
{
        struct disk *du;
        int     lunit = dkunit(bp->b_dev);
        int     s;

        /* valid unit, controller, and request?  */
        if (lunit >= NWD || bp->b_blkno < 0 || (du = wddrives[lunit]) == NULL
            || bp->b_bcount % DEV_BSIZE != 0) {

                bp->b_error = EINVAL;
                bp->b_flags |= B_ERROR;
                goto done;
        }
        s = splbio();
        bufqdisksort(&drive_queue[lunit], bp);

        if (wdutab[lunit].b_active == 0)
                wdustart(du);   /* start drive */
...
        splx(s);

        return;
}
```

But is it UNIX?

Linux *drivers/ide/ide-disk.c* :

```
static ide_startstop_t do_rw_disk (ide_drive_t *drive, struct request *rq, unsigned l
{
        if (IDE_CONTROL_REG)
                OUT_BYTE(drive->ctl,IDE_CONTROL_REG);
        OUT_BYTE(0x00, IDE_FEATURE_REG);
...
        if (rq->cmd == WRITE) {
                ide_startstop_t startstop;
                OUT_BYTE(drive->mult_count ? WIN_MULTWRITE : WIN_WRITE, IDE_COMMAND_R
                if (ide_wait_stat(&startstop, drive, DATA_READY, drive->bad_wstat, WA
                        printk(KERN_ERR "%s: no DRQ after issuing %s0, drive->name,
                                drive->mult_count ? "MULTWRITE" : "WRITE");
                        return startstop;
                }
                if (!drive->unmask)
                        __cli();        /* local CPU only */
...
}
```

But is it UNIX?

Seventh Edition *sys/h/buf.h* :

```
struct buf
{
        int     b_flags;                /* see defines below */
        struct  buf *b_forw;            /* headed by d_tab of conf.c */
        struct  buf *b_back;            /*  "  */
        struct  buf *av_forw;           /* position on free list, */
        struct  buf *av_back;           /*     if not BUSY*/
        dev_t   b_dev;                  /* major+minor device name */
        unsigned b_bcount;              /* transfer count */
        union {
            caddr_t b_addr;             /* low order core address */
            int *b_words;               /* words for clearing */
            struct filsys *b_filsys;    /* superblocks */
            struct dinode *b_dino;      /* ilist */
            daddr_t *b_daddr;           /* indirect block */
        } b_un;
        daddr_t b_blkno;                /* block # on device */
        char    b_xmem;                 /* high order core address */
        char    b_error;                /* returned after I/O */
        unsigned int b_resid;           /* words not transferred after error */
};
```

But is it UNIX?

FreeBSD *sys/sys/buf.h* :

```
struct buf {
        LIST_ENTRY(buf) b_hash;         /* Hash chain. */
        LIST_ENTRY(buf) b_vnbufs;       /* Buffer's associated vnode. */
        TAILQ_ENTRY(buf) b_freelist;    /* Free list position if not active. */
        TAILQ_ENTRY(buf) b_act;         /* Device driver queue when active. *new* */
        struct  proc *b_proc;           /* Associated proc; NULL if kernel. */
        long    b_flags;                /* B_* flags. */
        unsigned short b_qindex;        /* buffer queue index */
        unsigned char b_usecount;       /* buffer use count */
        int     b_error;                /* Errno value. */
        long    b_bufsize;              /* Allocated buffer size. */
        long    b_bcount;               /* Valid bytes in buffer. */
        long    b_resid;                /* Remaining I/O. */
        dev_t   b_dev;                  /* Device associated with buffer. */
        struct {
                caddr_t b_addr;         /* Memory, superblocks, indirect etc. */
        } b_un;
        caddr_t b_kvabase;              /* base kva for buffer */
        int     b_kvasize;              /* size of kva for buffer */
        void    *b_saveaddr;            /* Original b_addr for physio. */
        daddr_t b_lblkno;               /* Logical block number. */
        daddr_t b_blkno;                /* Underlying physical block number. */
...
};
```

But is it UNIX?

Linux *include/linux/fs.h* :

```
struct buffer_head {
        /* First cache line: */
        struct buffer_head *b_next;     /* Hash queue list */
        unsigned long b_blocknr;        /* block number */
        unsigned short b_size;          /* block size */
        unsigned short b_list;          /* List that this buffer appears */
        kdev_t b_dev;                   /* device (B_FREE = free) */

        atomic_t b_count;               /* users using this block */
        kdev_t b_rdev;                  /* Real device */
        unsigned long b_state;          /* buffer state bitmap (see above) */
        unsigned long b_flushtime;      /* Time when (dirty) buffer should be written

        struct buffer_head *b_next_free;/* lru/free list linkage */
        struct buffer_head *b_prev_free;/* doubly linked list of buffers */
        struct buffer_head *b_this_page;/* circular list of buffers in one page */
        struct buffer_head *b_reqnext;  /* request queue */

        struct buffer_head **b_pprev;   /* doubly linked list of hash-queue */
        char * b_data;                  /* pointer to data block */
        struct page *b_page;            /* the page this bh is mapped to */
        void (*b_end_io)(struct buffer_head *bh, int uptodate); /* I/O completion */
        void *b_private;                /* reserved for b_end_io */
...
};
```

Why such a secret?

Why such a secret?

- BSD developers are not commercially oriented.

Why such a secret?

- BSD developers are not commercially oriented.
- No companies to market BSD.

Why such a secret?

- BSD developers are not commercially oriented.
- No companies to market BSD.
- BSD is ‘‘for experts’’, not so easy to use.

Why such a secret?

- BSD developers are not commercially oriented.
- No companies to market BSD.
- BSD is ‘‘for experts’’, not so easy to use.
- BSD lawsuit scared people off.

The BSD wars

The BSD wars

- In 1992, UNIX Systems Laboratories sued BSDI, alleging copyright infringement.

The BSD wars

- In 1992, UNIX Systems Laboratories sued BSDI, alleging copyright infringement.
- The case was settled out of court in 1994.

The BSD wars

- In 1992, UNIX Systems Laboratories sued BSDI, alleging copyright infringement.
- The case was settled out of court in 1994.
- BSD may no longer be called UNIX.

The BSD wars

- In 1992, UNIX Systems Laboratories sued BSDI, alleging copyright infringement.
- The case was settled out of court in 1994.
- BSD may no longer be called UNIX.
- People are still afraid of litigation.

The BSD license

- Written late, after separation from UNIX.

The BSD license

- Written late, after separation from UNIX.
- The fewest restrictions of any free software license.

The BSD license

- Written late, after separation from UNIX.
- The fewest restrictions of any free software license.
- A bone of contention in the ‘‘UNIX wars’’.

The BSD license

- Written late, after separation from UNIX.
- The fewest restrictions of any free software license.
- A bone of contention in the ‘‘UNIX wars’’.
- A good choice for embedded systems.

The BSD license

Copyright (c) 1982, 1986, 1989, 1993
The Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The BSD license

The old version of the license contained an additional clause, to which rms objected:

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
   This product includes software developed by the University of California, Berkeley and its contributors.

Differences between BSDL and GPL

The BSD license is:

- Shorter.

Differences between BSDL and GPL

The BSD license is:

- Shorter.
- Not so restrictive.

Differences between BSDL and GPL

The BSD license is:

- Shorter.
- Not so restrictive.
- ‘‘Compatible’’ with GPL.

Differences between BSDL and GPL

The BSD license is:

- Shorter.
- Not so restrictive.
- ‘‘Compatible’’ with GPL.
- Less of a hassle for software developers.

Differences between BSDL and GPL

The BSD license is:

- Shorter.
- Not so restrictive.
- ‘‘Compatible’’ with GPL.
- Less of a hassle for software developers.
- Less suitable for releasing proprietary code.

What about free UNIX?

- UNIX was not free software.

What about free UNIX?

- UNIX was not free software.
- Attempts were made to make old versions of UNIX available under less restrictive licenses.

What about free UNIX?

- UNIX was not free software.
- Attempts were made to make old versions of UNIX available under less restrictive licenses.
- SCO created an ‘‘ancient UNIX’’ license for $100 US.

What about free UNIX?

- UNIX was not free software.
- Attempts were made to make old versions of UNIX available under less restrictive licenses.
- SCO created an ‘‘ancient UNIX’’ license for $100 US.
- Attempts were made to make old versions of UNIX available under less restrictive licenses.

What about free UNIX?

- UNIX was not free software.
- Attempts were made to make old versions of UNIX available under less restrictive licenses.
- SCO created an ‘‘ancient UNIX’’ license for $100 US.
- Attempts were made to make old versions of UNIX available under less restrictive licenses.
- Finally, in January 2002, UNIX was made available under a BSD license.

UNIX is free

Date: Wed, 23 Jan 2002 15:03:37 -0800
From: Dion Johnson <dionj@caldera.com>
To: wht@minnie.tuhs.org
Cc: dmr@bell-labs.com, ken@plan9.bell-labs.com, grog@lemis.com,
        John Terpstra <jht@caldera.com>, drew@caldera.com, maddog@li.org,
        evan@starnix.com, phatch@caldera.com, ransom@caldera.com
Subject: Liberal license for ancient UNIX sources

Dear Warren, and friends,

I'm happy to let you know that Caldera International has placed
the ancient UNIX releases (V1-7 and 32V) under a "BSD-style" license.
I've attached a PDF of the license letter hereto. Feel free to
propogate it as you see fit.

I apologize that this has taken so long. We do not have a well
regulated archive of these ancient releases, so we must depend
upon you UNIX enthusiasts, historians, and original authors to
help the community of interested parties figure out exactly what
is available, where, and how.
...
Anyway, here it is. Feel free to write to us if you want to
understand more about how/why Caldera International has released
this code, or you have any other comments that we should hear.

Sincerely,

Dion L. Johnson II - dionj@caldera.com
Product Manager and one of many open source enthusiasts in Caldera Intl.

Integration

- No ‘‘distributions’’, just three separate projects.

Integration

- No ‘‘distributions’’, just three separate projects.
- Each project has a single code base.

Integration

- No ‘‘distributions’’, just three separate projects.
- Each project has a single code base.
- Userland and kernel matched to each other.

Integration

- No ‘‘distributions’’, just three separate projects.
- Each project has a single code base.
- Userland and kernel matched to each other.
- Clear distinction between base system and add-ons.

Integration

- No ‘‘distributions’’, just three separate projects.
- Each project has a single code base.
- Userland and kernel matched to each other.
- Clear distinction between base system and add-ons.
- Man pages bundled with products.

Systems administration

- Motto: ‘‘Use standard tools’’.

Systems administration

- Motto: ‘‘Use standard tools’’.
- Nothing like SMIT, SAM, *linuxconf* or *YaST*.

Systems administration

- Motto: ‘‘Use standard tools’’.
- Nothing like SMIT, SAM, *linuxconf* or *YaST*.
- System-specific startup configuration defined in a single file, */etc/rc.conf*.

Systems administration

- Motto: ‘‘Use standard tools’’.
- Nothing like SMIT, SAM, *linuxconf* or *YaST*.
- System-specific startup configuration defined in a single file, */etc/rc.conf*.
- Default startup configuration defined in a single file, */etc/defaults/rc.conf*.

Systems administration

- Motto: ‘‘Use standard tools’’.
- Nothing like SMIT, SAM, *linuxconf* or *YaST*.
- System-specific startup configuration defined in a single file, */etc/rc.conf*.
- Default startup configuration defined in a single file, */etc/defaults/rc.conf*.
- Other */etc/rc** files perform specific functions, are not expected to change.

System startup: init

- *init* has not adopted the System V extensions which Linux also uses.

System startup: init

- *init* has not adopted the System V extensions which Linux also uses.
- No ‘‘run levels’’.

System startup: init

- *init* has not adopted the System V extensions which Linux also uses.
- No ‘‘run levels’’.
- Two states: single user mode, multi user mode.

System startup: init

- *init* has not adopted the System V extensions which Linux also uses.
- No ‘‘run levels’’.
- Two states: single user mode, multi user mode.
- Recently, NetBSD introduced a new configuration scheme which is being investigated by the other projects.

The UNIX File System

- Derived from the old UNIX file system round about 1982.

The UNIX File System

- Derived from the old UNIX file system round about 1982.
- Originally called ‘‘Fast File System’’ (ffs).

The UNIX File System

- Derived from the old UNIX file system round about 1982.
- Originally called ‘‘Fast File System’’ (ffs).
- Adopted in UNIX System V.4 as ‘‘UNIX File System’’ (ufs).

The UNIX File System

- Derived from the old UNIX file system round about 1982.
- Originally called ‘‘Fast File System’’ (ffs).
- Adopted in UNIX System V.4 as ‘‘UNIX File System’’ (ufs).
- Uses 64 bit byte pointers, 32 bit block pointers.

The UNIX File System

- Derived from the old UNIX file system round about 1982.
- Originally called ‘‘Fast File System’’ (ffs).
- Adopted in UNIX System V.4 as ‘‘UNIX File System’’ (ufs).
- Uses 64 bit byte pointers, 32 bit block pointers.
- Limited to 1 TB file system and file size.

The UNIX File System

- File partitioning BSD-specific, does not need a PC partition table.

The UNIX File System

- File partitioning BSD-specific, does not need a PC partition table.
- In conjunction with PC partition table, requires only one partition.

The UNIX File System

- File partitioning BSD-specific, does not need a PC partition table.
- In conjunction with PC partition table, requires only one partition.
- No journalling; uses ‘‘soft updates’’ and checkpointing instead for better performance than journalling.

The UNIX File System

- File partitioning BSD-specific, does not need a PC partition table.
- In conjunction with PC partition table, requires only one partition.
- No journalling; uses ‘‘soft updates’’ and checkpointing instead for better performance than journalling.
- Has a reputation for stability: c’t magazine tried 170 hot shutdowns but didn’t break it.

Userland tools

- Many tools derived from the original BSD code, for example *csh*.

Userland tools

- Many tools derived from the original BSD code, for example *csh*.
- No AT&T tools. *vi* was based on *ex*, so it couldn't be used.

Userland tools

- Many tools derived from the original BSD code, for example *csh*.
- No AT&T tools. *vi* was based on *ex*, so it couldn't be used.
- Most of the missing tools were replaced with GNU tools.

The Ports Collection

- Collection of third-party software.

The Ports Collection

- Collection of third-party software.
- Does *not* include base system functionality.

The Ports Collection

- Collection of third-party software.
- Does *not* include base system functionality.
- Some ports duplicate base system functionality.

The Ports Collection

- Collection of third-party software.
- Does *not* include base system functionality.
- Some ports duplicate base system functionality.
- Typically built from source (‘‘port’’).

The Ports Collection

- Collection of third-party software.
- Does *not* include base system functionality.
- Some ports duplicate base system functionality.
- Typically built from source (‘‘port’’).
- May be supplied as a binary ‘‘package’’.

The Ports Collection

- Collection of third-party software.
- Does *not* include base system functionality.
- Some ports duplicate base system functionality.
- Typically built from source (‘‘port’’).
- May be supplied as a binary ‘‘package’’.
- NetBSD has different terminology: ports are called ‘‘packages’’, and packages are called ‘‘pre-compiled packages’’.

The Ports Collection

- Ports are stored in */usr/ports*.

The Ports Collection

- Ports are stored in */usr/ports*.
- Top-level */usr/ports* directory contains subdirectories:

```
archivers       deskutils       hebrew          new             ukrainian
astro           devel           irc             news            vietnamese
audio           distfiles       japanese        palm            www
benchmarks      editors         java            picobsd         x11
biology         emulators       korean          print           x11-clocks
cad             foo             lang            russian         x11-fm
chinese         french          mail            science         x11-fonts
comms           ftp             math            security        x11-servers
converters      games           mbone           shells          x11-toolkits
cross           german          misc            sysutils        x11-wm
databases       graphics        net             textproc        xc
```

The Ports Collection

Each directory contains subdirectories for individual ports:

```
$ ls java
CVS               guavac          jdbcpool        jfc             mmake
Makefile          infobus         jde             jikes           netrexx
bluej             jad             jde-emacs20     jlex            openjit
bouncycastle      jaf             jdk             jlint           perltools
bsh               jakarta-oro     jdk-tutorial    jre             pkg
bugseeker         jakarta-regexp  jdk11-doc       jsdk            shujit
bugseeker-demo    janosvm         jdk12-beta      junit           tya
collections       java-cup        jdk12-doc       kaffe           xalan-j
cos               javamail        jdk13           linux-ibm-jdk13
cryptix-jce       javavmwrapper   jdk13-doc       linux-jdk
forte             javel           jdk14-doc       linux-jdk13
gnu-regexp        jce-aba         jdom            linux-jdk14
```

Port contents

- A *port* is the framework from which a package is created.

Port contents

- A *port* is the framework from which a package is created.
- Functionally, a port is equivalent to an SRPM without the source tarball.

Port contents

- A *port* is the framework from which a package is created.
- Functionally, a port is equivalent to an SRPM without the source tarball.
- Writing a port is equivalent to writing an RPM spec file.

Port contents

- A *port* is the framework from which a package is created.
- Functionally, a port is equivalent to an SRPM without the source tarball.
- Writing a port is equivalent to writing an RPM spec file.
- A port skeleton consists of a small directory tree with several files.

Port contents

- A *port* is the framework from which a package is created.
- Functionally, a port is equivalent to an SRPM without the source tarball.
- Writing a port is equivalent to writing an RPM spec file.
- A port skeleton consists of a small directory tree with several files.
- No special tools (*rpm*, *apt*) needed to build ports.

Port contents

- A *port* is the framework from which a package is created.
- Functionally, a port is equivalent to an SRPM without the source tarball.
- Writing a port is equivalent to writing an RPM spec file.
- A port skeleton consists of a small directory tree with several files.
- No special tools (*rpm*, *apt*) needed to build ports.
- All done with standard UNIX tools: *make*, *patch*, etc.

Port contents

- A *port* is the framework from which a package is created.
- Functionally, a port is equivalent to an SRPM without the source tarball.
- Writing a port is equivalent to writing an RPM spec file.
- A port skeleton consists of a small directory tree with several files.
- No special tools (*rpm*, *apt*) needed to build ports.
- All done with standard UNIX tools: *make*, *patch*, etc.
- Most use BSD *make*, which is very different from GNU *make*.

The Ports Collection

```
# cd /usr/ports/x11/xlogout/
# ls -RC
CVS             distinfo        pkg-comment     pkg-plist
Makefile        files           pkg-descr

./CVS:
Entries         Repository      Root

./files:
CVS             patch-aa

./files/CVS:
Entries         Repository      Root
```

Port Makefile

- Most of Makefile included from */usr/ports/mk/bsd.port.mk*.

Port Makefile

- Most of Makefile included from */usr/ports/mk/bsd.port.mk*.
- Only a few keywords required in port *Makefile*.

Port Makefile

- Most of Makefile included from */usr/ports/mk/bsd.port.mk*.
- Only a few keywords required in port *Makefile*.
- Targets available for each step of porting.

Port Makefile

- Most of Makefile included from */usr/ports/mk/bsd.port.mk*.
- Only a few keywords required in port *Makefile*.
- Targets available for each step of porting.
- Target `package` builds a package.

Port Makefile

```
# New ports collection makefile for:   xlogout
# Date created:               1998-11-06
# Whom:                       Christian Weisgerber <naddy@mips.rhein-neckar.de>
#
# $FreeBSD: ports/x11/xlogout/Makefile,v 1.6 2001/11/03 22:22:34 naddy Exp $
#

PORTNAME=       xlogout
PORTVERSION=    1.1
CATEGORIES=     x11
MASTER_SITES=   ftp://ftp.tu-darmstadt.de/pub/X11/other/
EXTRACT_SUFX=   .tar.Z

MAINTAINER=     naddy@FreeBSD.org

WRKSRC=         ${WRKDIR}/xlogout
USE_IMAKE=      yes
MAN1=           xlogout.1
```

Building ports

Nothing to it:

```
# cd /usr/ports/x11/xlogout
# make install
```

Building ports

Building a port performs the following:

- Checks whether the distribution source tarball is available locally. If not, it fetches it from the net.

Building ports

Building a port performs the following:

- Checks whether the distribution source tarball is available locally. If not, it fetches it from the net.
- Verifies the checksums.

Building ports

Building a port performs the following:

- Checks whether the distribution source tarball is available locally. If not, it fetches it from the net.
- Verifies the checksums.
- Extract the source tree into a working directory.

Building ports

Building a port performs the following:

- Checks whether the distribution source tarball is available locally. If not, it fetches it from the net.
- Verifies the checksums.
- Extract the source tree into a working directory.
- Applies any distribution patches.

Building ports

- Applies any port patches that may be required to adapt the target software to BSD or the conventions of the Ports Collection.

Building ports

- Applies any port patches that may be required to adapt the target software to BSD or the conventions of the Ports Collection.
- Configures the port (e.g. with *configure*).

Building ports

- Applies any port patches that may be required to adapt the target software to BSD or the conventions of the Ports Collection.
- Configures the port (e.g. with *configure*).
- Builds (compiles) the program.

Building ports

- Applies any port patches that may be required to adapt the target software to BSD or the conventions of the Ports Collection.
- Configures the port (e.g. with *configure*).
- Builds (compiles) the program.
- Installs the program.

Building a port

```
# make install
>> xlogout-1.1.tar.Z doesn't seem to exist on this system.
>> Attempting to fetch from ftp://ftp2.de.freebsd.org/pub/FreeBSD/ports/distfiles/.
Receiving xlogout-1.1.tar.Z (3466 bytes): 100%
3466 bytes transferred in 1.9 seconds  (1.75 Kbytes/s)
===>  Extracting for xlogout-1.1
>> Checksum OK for xlogout-1.1.tar.Z.
===>   xlogout-1.1 depends on executable: imake - found
===>   xlogout-1.1 depends on shared library: X11.6 - found
===>  Patching for xlogout-1.1
===>  Applying FreeBSD patches for xlogout-1.1
===>  Configuring for xlogout-1.1
imake -DUseInstalled -I/usr/X11R6/lib/X11/config
make Makefiles
make includes
make depend
rm -f .depend
gccmakedep -f-  --   -I/usr/X11R6/include   -DCSRG_BASED   -DFUNCPROTO=15
 -DNARROWPROTO    -- xlogout.c > .depend
===>  Building for xlogout-1.1
cc -Os -pipe      -I/usr/X11R6/include   -DCSRG_BASED   -DFUNCPROTO=15
 -DNARROWPROTO   -c xlogout.c
xlogout.c: In function 'main':
xlogout.c:52: warning: return type of 'main' is not 'int'
```

Building a port

```
rm -f xlogout
cc -o xlogout       -L/usr/X11R6/lib xlogout.o -lXaw -lXmu -lXt -lSM -lICE
 -lXext -lX11    -lxpg4  -Wl,-rpath,/usr/X11R6/lib
/usr/X11R6/lib/libXaw.so: warning: tmpnam() possibly used unsafely; consider
using mkstemp()
===>  Installing for xlogout-1.1
===>   xlogout-1.1 depends on shared library: X11.6 - found
/usr/bin/install -c -s  xlogout /usr/X11R6/bin/xlogout
/usr/bin/install -c -m 0444 XLogout.ad /usr/X11R6/lib/X11/app-defaults/XLogout
install in . done
rm -f /usr/X11R6/man/man1/xlogout.1*
/usr/bin/install -c -m 0444 xlogout.man  /usr/X11R6/man/man1/xlogout.1
gzip -n /usr/X11R6/man/man1/xlogout.1
install.man in . done
===>   Generating temporary packing list
===>   Registering installation for xlogout-1.1
===>  Cleaning for XFree86-3.3.6
===>  Cleaning for xlogout-1.1
```

The team

- The *core team* or *core group* controls the project with a loose rein.

The team

- The *core team* or *core group* controls the project with a loose rein.
- *Committers* are developers with direct write access to the source tree.

The team

- The *core team* or *core group* controls the project with a loose rein.
- *Committers* are developers with direct write access to the source tree.
- *Contributors* are developers without direct write access to the source tree.

The team

- The *core team* or *core group* controls the project with a loose rein.
- *Committers* are developers with direct write access to the source tree.
- *Contributors* are developers without direct write access to the source tree.
- Many committers work on more than one BSD project.

BSD releases

- Continuous development in the `CURRENT` branch, for developers only.

BSD releases

- Continuous development in the CURRENT branch, for developers only.
- Regular releases, called RELEASE.

BSD releases

- Continuous development in the CURRENT branch, for developers only.
- Regular releases, called RELEASE.
- Separate bug fixes to releases in the STABLE branch.

BSD development model

- The BSD projects maintain a single copy of the system sources for all releases.

BSD development model

- The BSD projects maintain a single copy of the system sources for all releases.
- Sources maintained by CVS, allowing easy updating and backout of changes.

BSD development model

- The BSD projects maintain a single copy of the system sources for all releases.
- Sources maintained by CVS, allowing easy updating and backout of changes.
- Sources include the complete operating system, not just the kernel.

BSD development model

- The BSD projects maintain a single copy of the system sources for all releases.
- Sources maintained by CVS, allowing easy updating and backout of changes.
- Sources include the complete operating system, not just the kernel.
- CVS allows access to any version of any file in a single tree.

BSD development model

- The BSD projects maintain a single copy of the system sources for all releases.
- Sources maintained by CVS, allowing easy updating and backout of changes.
- Sources include the complete operating system, not just the kernel.
- CVS allows access to any version of any file in a single tree.
- Clear development model: complete project history is available.

A CVS log: *sys/kern/kern_exec.c*

```
RCS file: kern_exec.c,v
Working file: kern_exec.c
head: 1.110
branch:
locks: strict
access list:
symbolic names:
        RELENG_4_0_0_RELEASE: 1.107
        RELENG_4: 1.107.0.2
        RELENG_4_BP: 1.107
        RELENG_3_4_0_RELEASE: 1.93.2.3
        RELENG_3_3_0_RELEASE: 1.93.2.3
        RELENG_3_2_PAO: 1.93.2.1.0.2
        RELENG_3_2_PAO_BP: 1.93.2.1
        RELENG_3_2_0_RELEASE: 1.93.2.1
        POST_VFS_BIO_NFS_PATCH: 1.99
        PRE_VFS_BIO_NFS_PATCH: 1.99
        POST_SMP_VMSHARE: 1.99
        PRE_SMP_VMSHARE: 1.99
        POST_NEWBUS: 1.97
        PRE_NEWBUS: 1.97
        RELENG_3_1_0_RELEASE: 1.93
        RELENG_3: 1.93.0.2
        RELENG_3_BP: 1.93
        RELENG_2_2_8_RELEASE: 1.47.2.12
```

A CVS log (2)

```
total revisions: 138;   selected revisions: 138
description:
----------------------------
revision 1.110
date: 2000/04/26 20:58:39;  author: dillon;  state: Exp;  lines: +38 -29
    Fix #! script exec under linux emulation.  If a script is exec'd from a
    program running under linux emulation, the script binary is checked for
    in /compat/linux first.  Without this patch the wrong script binary
    (i.e. the FreeBSD binary) will be run instead of the linux binary.
    For example, #!/bin/sh, thus breaking out of linux compatibility mode.

    This solves a number of problems people have had installing linux
    software on FreeBSD boxes.
----------------------------
revision 1.109
date: 2000/04/18 15:15:18;  author: phk;  state: Exp;  lines: +1 -2
Remove unneeded <sys/buf.h> includes.

Due to some interesting cpp tricks in lockmgr, the LINT kernel shrinks
by 924 bytes.
----------------------------
revision 1.108
date: 2000/04/16 18:53:09;  author: jlemon;  state: Exp;  lines: +6 -1
Introduce kqueue() and kevent(), a kernel event notification facility.
```

Another CVS log

```
revision 1.152
date: 1997/10/06 09:58:11;  author: jkh;  state: Exp;  lines: +41 -13
Hooboy!

Did I ever spam this file good with that last commit.  Despite 3
reviewers, we still managed to revoke the eBones fixes, TCL 8.0 support,
libvgl and a host of other new things from this file in the process of
parallelizing the Makefile.  DOH!   I think we need more pointy hats - this
particular incident is worthy of a small children's birthday party's worth of
pointy hats. ;-)

I certainly intend to take more care with the processing of aged diffs
in the future, even if it does mean reading through 20K's worth of them.
I might also be a bit more careful about asking for more up-to-date changes
before looking at them. ;)
```

Project coordination

- Communication primarily by mail.

Project coordination

- Communication primarily by mail.
- Many special-purpose mailing lists.

Project coordination

- Communication primarily by mail.
- Many special-purpose mailing lists.
- Some developers communicate in real time with *irc*.

Project coordination

- Communication primarily by mail.
- Many special-purpose mailing lists.
- Some developers communicate in real time with *irc*.
- Source update with *CVSup* or *sup*.

The NetBSD project

- Founded in April 1993.

Goals:

- Architecturally clean.
- Highly portable.
- Highly interoperable.
- State-of-the-art security.
- Core group of 5 members.
- Over 150 committers with direct access to source tree.
- How many users?

The FreeBSD project

- Founded in December 1993.
- Core team of 9 members, elected by the committers.
- Over 250 committers with direct access to source tree.
- Over 2 million users.
- 4 releases per year.
- 5,000 applications in the ‘‘Ports Collection’’.
- Release 4.4 in September 2001.

The OpenBSD project

- Derived from NetBSD in 1995.
- ‘‘Benevolent Dictatorship’’ under leader Theo de Raadt.
- Approximately 115 committers with direct access to the source tree.
- How many users?

OpenBSD

- Portability.
- Standardization.
- Correctness.
- Proactive security.
- Integrated cryptography.
- Largely developed by non-Americans.
- Has proven to be a testbed for ‘‘cryptography inside an operating system’’.

Who uses BSD?

Who uses BSD?

- Yahoo! (FreeBSD)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)
- IBM (Whistle InterJet, FreeBSD)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)
- IBM (Whistle InterJet, FreeBSD)
- International Space Station (NetBSD)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)
- IBM (Whistle InterJet, FreeBSD)
- International Space Station (NetBSD)
- (***Censored***) (OpenBSD)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)
- IBM (Whistle InterJet, FreeBSD)
- International Space Station (NetBSD)
- (***Censored***) (OpenBSD)
- Telstra Internet (FreeBSD)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)
- IBM (Whistle InterJet, FreeBSD)
- International Space Station (NetBSD)
- (***Censored***) (OpenBSD)
- Telstra Internet (FreeBSD)
- Apple computer (MacOS X)

Who uses BSD?

- Yahoo! (FreeBSD)
- Hotmail (FreeBSD)
- IBM (Whistle InterJet, FreeBSD)
- International Space Station (NetBSD)
- (***Censored***) (OpenBSD)
- Telstra Internet (FreeBSD)
- Apple computer (MacOS X)
- Who knows? (other embedded systems)

Apple

- Apple MacOS X based on BSD.

Apple

- Apple MacOS X based on BSD.
- Apple's low-level operating system (Darwin) is free software.

Apple

- Apple MacOS X based on BSD.
- Apple's low-level operating system (Darwin) is free software.
- Apple developers have commit access to FreeBSD source tree.

Apple

- Apple MacOS X based on BSD.
- Apple's low-level operating system (Darwin) is free software.
- Apple developers have commit access to FreeBSD source tree.
- Changes in Darwin returned to FreeBSD.

Apple

- Apple MacOS X based on BSD.
- Apple's low-level operating system (Darwin) is free software.
- Apple developers have commit access to FreeBSD source tree.
- Changes in Darwin returned to FreeBSD.
- FreeBSD changes (SMP) planned for inclusion in MacOS X.

Security

- No viruses.

Security

- No viruses.
- Include *OpenSSL* and *OpenSSH*.

Security

- No viruses.
- Include *OpenSSL* and *OpenSSH*.
- IPSec (OpenBSD was the first operating system to include it).

Security

- No viruses.
- Include *OpenSSL* and *OpenSSH*.
- IPSec (OpenBSD was the first operating system to include it).
- Each project has a security team.

Security

- No viruses.
- Include *OpenSSL* and *OpenSSH*.
- IPSec (OpenBSD was the first operating system to include it).
- Each project has a security team.
- Regular security advisories.

Desktop support

- BSD considered user-unfriendly.

Desktop support

- BSD considered user-unfriendly.
- But it uses *exactly* the same desktop software as Linux.

Desktop support

- BSD considered user-unfriendly.
- But it uses *exactly* the same desktop software as Linux.
- Linux applications software runs on BSD unchanged.

Desktop support

- BSD considered user-unfriendly.
- But it uses *exactly* the same desktop software as Linux.
- Linux applications software runs on BSD unchanged.
- Apple's MacOS X is based on BSD.

Which BSD?

Which BSD?

- Difficult to put the projects into a box.

Which BSD?

- Difficult to put the projects into a box.
- All three projects produce high performance, portable, secure systems, but the slogans say...

Which BSD?

- Difficult to put the projects into a box.
- All three projects produce high performance, portable, secure systems, but the slogans say...
- FreeBSD for performance: ‘‘The power to serve’’.

Which BSD?

- Difficult to put the projects into a box.
- All three projects produce high performance, portable, secure systems, but the slogans say...
- FreeBSD for performance: ‘‘The power to serve’’.
- NetBSD for portability: ‘‘Of course it runs NetBSD’’.

Which BSD?

- Difficult to put the projects into a box.
- All three projects produce high performance, portable, secure systems, but the slogans say...
- FreeBSD for performance: ‘‘The power to serve’’.
- NetBSD for portability: ‘‘Of course it runs NetBSD’’.
- OpenBSD for security: ‘‘Four years without a remote hole in the default install’’.

Current releases

- FreeBSD 4.4 was released in September 2001.
- FreeBSD 4.4 was released in January 2002.
- NetBSD 1.5.1 was released in July 2001.
- NetBSD 1.5.2 was released in September 2001.
- OpenBSD 2.9 was released in June 2001.
- OpenBSD 3.0 was released in December 2001.

Mailing lists

- `FreeBSD-questions@FreeBSD.org`
- `NetBSD-help@NetBSD.org`
- `misc@OpenBSD.org`
- See web sites for signup instructions.

Who supports BSD?

- FreeBSD Services support FreeBSD.

Who supports BSD?

- FreeBSD Services support FreeBSD.
- FreeBSD Mall supports FreeBSD.

Who supports BSD?

- FreeBSD Services support FreeBSD.
- FreeBSD Mall supports FreeBSD.
- Wasabi Systems support NetBSD.

Who supports BSD?

- FreeBSD Services support FreeBSD.
- FreeBSD Mall supports FreeBSD.
- Wasabi Systems support NetBSD.
- In Australia, Tellurian supports BSD.

Who supports BSD?

- FreeBSD Services support FreeBSD.
- FreeBSD Mall supports FreeBSD.
- Wasabi Systems support NetBSD.
- In Australia, Tellurian supports BSD.
- Many individuals.

Why? Is BSD better than Linux?

- For the end user, there's little difference.

Why? Is BSD better than Linux?

- For the end user, there's little difference.
- Changing in either direction takes an effort.

Why? Is BSD better than Linux?

- For the end user, there's little difference.
- Changing in either direction takes an effort.
- If you like what you have, stick with it.

Acknowledgements

The free software ethic says ‘‘don’t reinvent the wheel’’.

Acknowledgements

The free software ethic says ‘‘don’t reinvent the wheel’’.

Thanks for contributions from:

- Jordan Hubbard <jkh@FreeBSD.org>
- Luke Mewburn <lukem@NetBSD.org>
- Theo de Raadt <deraadt@cvs.OpenBSD.org>
- Christian Weisgerber <naddy@mips.inka.de>

For more information

- These slides are available at `http://www.lemis.com/linux.conf.au.pdf`
- `http://www.FreeBSD.org/`
- `http://www.NetBSD.org/`
- `http://www.OpenBSD.org/`
- `http://www.apple.com/`
- `http://www.bsdi.com/`
- `http://www.wasabi.com/`
- `http://www.FreeBSD-services.com/`
- `http://www.FreeBSDmall.com/`